

# IBM System/3

From Wikipedia, the free encyclopedia

The **IBM System/3** (introduced 1969 discontinued 1985) was a low-end business computer aimed at new customers and organizations that still used IBM 1400 series computers or unit record equipment. It was the first member of what IBM refers to as their "mid range" line and introduced the RPG programming language for which it was essentially an appliance for small business applications.

## Contents

- 1 Overview
- 2 Instruction set
- 3 Computer system
- 4 Legacy
- 5 Quirks
- 6 Chronology
- 7 References
- 8 Books by System/3 Authors
- 9 External links

## Overview

It featured a new punched card format that was smaller and stored 96 characters. Instead of the rectangular punches in the classic IBM card, the new cards had tiny (1 mm), circular holes much like paper tape. Data was stored in six-bit binary-coded decimal code, with three rows of 32 characters each, or 8-bit EBCDIC, with the two extra holes located in the top rows. Cards had room for 128 printed characters in four rows of 32 characters each. IBM System/370s with a proper card reader could also process the new cards.

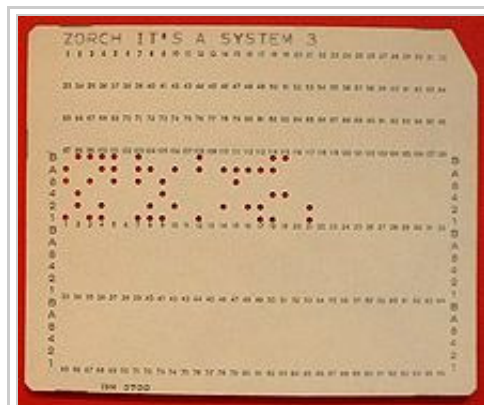


System 3 model 10D

Many of the original System/3 model 10 units were shipped *only* with card reader/sorter/punch I/O and a printer. There were two models of card reader. The most common was known as the MFCU ("multifunction card unit") which read, punched, and sorted the new 96 columns cards. The second, less common, card unit was the MFCM ("multifunction card machine") which performed the same operations, only with the more common 80 column cards. The MFCM was provided for larger companies that already had an investment in the 80 column card equipment. The printers available were multiple models of the existing IBM 1403 printer line, which had been originally built for the IBM 1401 computer. In a card-only system, the RPG/II compiler was supplied as two phases. The first phase would be booted from one input hopper of the MFCU, and the source would then be read following the compiler. An intermediate form was punched on cards, which were then read by the second phase of the compiler. An executable program deck



IBM System/3 with keypunch on display at the Olmsted County Historical Society Museum in Rochester, Minnesota.



96 column punched card.

was then punched. This executable could then be booted ("IPL'ed", for "Initial Program Load") to perform the processing desired. This process could require more than an hour for a significant sized program.

For mass storage, the System/3 used a single-platter disk, roughly the size of a large pizza; initially each platter held 2.5 MB of data. Standard configuration for storage was one or two fixed disks, each in a separate pull-out drawer, which typically held the operating system and user-developed programs. Additionally, each fixed disc could have a removable cartridge disk attached; these typically contained the data-files associated with various applications, for example Payroll, and users frequently had a number of them. Thus the low-end systems could support a maximum of 10 MB of online storage (two fixed, 2 removable), although in practice this was very expensive and not always common.

Offline storage was available with the purchase of an external tape drive which read and wrote standard IBM tape content.

System printing was typically via line printers or bi-directional dot matrix printers. A modified selectric typewriter was often used as a console.

The System/3 came standard with a RPG/II compiler, and used a simple job control language called OCL, which was superficially similar to the famous (or infamous) Job Control Language.

The System/3 and successor models (System/32, System/34, System/36 and System/38, then the AS/400 and iSeries) are generally referred to as minicomputers or in IBM terminology "midrange systems"—in contrast to IBM's more traditional large mainframes.

## Instruction set

One of the more interesting things about the System/3 was its instruction set. This was optimised for two key aspects of the product: limited availability of main memory, and the RPG programming language.

The original S/3 (models 10 and then 6, 8 & 12) had 29 instructions, all occupying between 3 and 6 bytes (24 to 48 bits). The first 4 bits conveyed a lot of information: "1111" meant this was an instruction without operands, known as a command. e.g. Start I/O (the I/O op being defined by previously loaded I/O registers). "11xx" and "xx11" meant a 1-operand instruction, such as a Branch. If xx was 00 the operand was addressed by its full 16-bit address. xx=01 or 10 meant base-displacement addressing was used, using index register 1 or 2 respectively. A base address would previously have been loaded into one of the two index registers and the instruction contained the displacement of up to just 256 bytes (8 bits of addressing).

Other patterns for this first half-byte indicated a 2-operand instruction. "0000" meant both operands were addressed by their direct 16-bit address. "0100": operand 1 uses reg 1 as its base; operand 2 uses direct addressing. "0110": operand 1 uses reg 1 as its base, operand 2 uses reg 2. And so on.

The remaining 4 bits of the first byte further defined the instruction. This structure meant that there was the capability to have up to 64 operations in all: 16 commands (though there were never more than five across the whole product range); 16 1-operand instructions starting with 11xx; 16 1-operand instructions starting with xx11; 16 2-operand instructions.

As well as the two index registers already mentioned (referred to as 1 and 2, or binary 01 and 10) there were other registers. "Reg 4" (0100) was the instruction address register (IAR) which pointed at the current instruction. "Reg 8" (1000) was the address recall register (ARR), set by certain instructions. Among these was the conditional branch (mnemonic BC) which used it to point to the byte immediately following the branch operation. For IBM mainframe folks this means that the S/3 branch could be likened to a conditional BALR (branch and link register). Very useful when branching to a sub-routine, and returning after it had processed.

Finally, "Reg 16" (00010000) was the program status register, holding such things as the results of a compare instruction. Note that registers were used only for addressing and program status, not for arithmetic.

The arithmetic instructions provided among the 29 instructions were binary add/subtract (provided to help manipulate addresses) and decimal add/subtract. Multiplication and division were not provided for by the standard hardware, and had to be handled by software routines. There was no floating point provision at all. All this continued to be true even with the later and generally more sophisticated Systems/34 and 36.

All the above got more complicated with the System/3 model 15, and the Systems/34 and 36. Though still using 16-bit addressing, all these systems could support well over 64K of main storage (up to 512K and theoretically more), so address translation was used to swap from one 64K address space to another. Address Translation Registers were set to define the actual address space in use at any one time, their contents being concatenated with the 16-bit address used by a program to produce a real address. These "ATRs" were privileged, available only to the operating system. (Just as well, or chaos would have ensued.)

As an aside, the original S/3 model 10 (and the later model 12) had an optional crude form of multi-programming called the Dual Program Feature. This provided no more main memory addressing, but gave two sets of registers and instructions which flipped from one "program level" to the other. The standard I/O instructions were also modified to flip when an I/O was started. It was possible to manipulate the other program level's instruction address register, so you could play tricks on the machine. All harmless fun!

So far, only the first byte of the instruction has been explained here. The next ("Q") byte was generally a qualifier, such as specifying the number of bytes to be moved in a move characters op or the condition to test for in a Branch. A couple of instructions used this byte for a 1-byte "immediate" operand. The remaining byte(s) were for the displacement(s) or address(es) for operands, or the details for some commands.

Here's an example to show all this at work: a simple command, Conditional Jump, a special type of conditional branch (forward only, up to 256 bytes) suitable mainly for jumping over short blocks of code: Op code byte= F2 (this is in hexadecimal, Hex F is binary 1111, Hex 2 (0010) defines the op); Q byte= 00000001 specifies that we "jump" if the condition register has the "equal" bit on; Operand= 00011000: if the condition is met we jump forward 24 bytes.

As previously stated, this instruction set was optimised for two aspects of the System/3 family. The first was the small amount of memory available, the second was the RPG programming language. Those who know anything about early RPG will know about indicators—binary switches used to control program flow. Over 100 of these were available to the programmer and they were well used. (Less so in more recent practice). By using the instruction formats explained above, many of the indicator-oriented operations could be fit into just 3 bytes. For example... a line of RPG might test an indicator for "On": 3 bytes for a "Test Bits On" op; then 3 bytes for a Jump, as previously described, and useful to the RPG compiler. Saving the odd byte here and there was good when you had only 64K to play with—and, on the S/3 itself, that had to include the operating system (which grew to about 20K on the model 10 with the introduction of the "Communication Control Program", CCP).

## Computer system

The IBM System/3 was announced as a computer system that initially consisted of <sup>[1]</sup>

- IBM 5410 Model 10 Central Processing Unit
- IBM 5424 Multi Functional Card Unit (MFCU)
- IBM 5203 Line Printer
- IBM 5444 Disk Storage
- IBM 5471 Printer Keyboard
- IBM 5475 Data Entry Keyboard

- IBM 5496 Data Recorder, a keypunch machine with print and verify functions
- IBM 5486 Card Sorter

The IBM 96-column card, instead of the regular IBM 80-column card, was used.

## Legacy

The single-user System/32 had a very different, 16-bit word oriented processor. It emulated the System/3 instruction set in software (rather slowly).

The later IBM System/34 and IBM System/36 partly inherited the architecture from System/32. They had two processors: a Control Storage Processor (CSP), same as in System/32, which handled most supervisor and input/output operations, and a Main Storage Processor (MSP), a re-implementation of the System/3 model 15 processor, used to run System/3 applications. So, those systems may be thought of as System/32 with added "hardware emulation" of System/3.

Although the IBM System/38 and its successor the AS/400 and iSeries filled the same market niche, they used a radically different architecture, based on the failed IBM Future Systems project.

## Quirks

The System/3 had no provision for halting a process once it had started to run. For instance, if a compile failed because of an error on the very first page, you had to wait for a sometimes voluminous compile listing to print in its entirety. One trick learned by programmers was that you could reach under the printer and jostle the paper discharge chute. This would cause the machine to halt and an error code P3 would be displayed on the console. You could then dial in the code FF to abort the run.

The System/3 also had no audible warning device, so a program could halt and unless the operator was looking at the console and noticed the status lights showed the Error Code he often did not realize that an event had occurred and the machine was not operating.

## Chronology

- 1969 - IBM 5410, or System/3 Model 10, introduced
- 1970 - IBM 5406, or System/3 Model 6, introduced
- 1973 - IBM 5415, or System/3 Model 15, introduced
- 1974 - IBM 5408, or System/3 Model 8, introduced
- 1975 - IBM 5412, or System/3 Model 12, introduced

The System/3 Operator Console Facility (OCF) consisted of either a IBM selectric typewriter interfaces into the computer, or a special purpose IBM 3270 display. Within the OCF, there was capability to 'cancel' processes and/or tasks that were running, including either partition (P1 or P2). The system could only run two programs simultaneously, except for the model 15 or systems running the Communication Control program, CCP.

## References

1. ^ IBM Field Engineering Announcement: IBM System/3 (<http://ed-thelen.org/comp-hist/IBM-ProdAnn/syst3.pdf>)

## Books by System/3 Authors

- "Everything You Always Wanted to Know About the System/3 But Nobody Told You" by Charlie Massoglia
- "System/3 Disk Sort as a Programming Language" by Charlie Massoglia

## External links

- "IBM System/3" ([http://www-03.ibm.com/ibm/history/exhibits/rochester/rochester\\_4008.html](http://www-03.ibm.com/ibm/history/exhibits/rochester/rochester_4008.html)) . IBM Archives. [http://www-03.ibm.com/ibm/history/exhibits/rochester/rochester\\_4008.html](http://www-03.ibm.com/ibm/history/exhibits/rochester/rochester_4008.html). Retrieved 2006-05-29.
- A System/3 under restoration at the CoreStore (<http://www.corestore.org/3.htm>)
- IBM System/3 website (<http://www.ibmssystem3.nl/>)

Retrieved from "[http://en.wikipedia.org/wiki/IBM\\_System/3](http://en.wikipedia.org/wiki/IBM_System/3)"

Categories: IBM minicomputers | Minicomputer stubs

---

- This page was last modified on 23 June 2010 at 22:04.
- Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. See Terms of Use for details.

Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.
- Privacy policy
- About Wikipedia
- Disclaimers